IV. Conclusion

We have shown that in addition to the original pole radii, the original pole angles are also involved in determining the minimum delay which can assure stability when using the clustered look-ahead or time-domain pipelining technique. Furthermore, for complex pole pairs of relatively large magnitude, the occurrence of stability pockets increases as $\cos(\phi) \rightarrow 0$, thus making the clustered look-ahead pipelining technique a more viable alternative than was once thought. In addition, we note that decomposition of high-order filters into lower order subsections can sometimes reduce the total computational complexity.

By exploiting this knowledge, drastically reduced hardware implementations are possible for stable clustered look-ahead pipelined IIR digital filters when compared to the scattered look-ahead technique. Furthermore, our technique guarantees we will obtain the minimum number of superfluous poles possible while maintaining exactly n multipliers in the denominator of a clustered look-ahead pipelined IIR filter [11]. Thus in many cases, we achieve the minimum number of superfluous poles with fewer multipliers than the minimum order augmentation technique of Lim and Liu [1] due to the reduced hardware expense of incremental augmentation.

References

[1] Y. Lim and B. Liu, "Pipelined recursive filter with minimum order augmentation," *IEEE Trans. Signal Process.*, vol. 40, pp. 1643–1651, July 1992.

[2] B. Sinha, "Pipelined finite state machine architecture applied to digital filters," Ph.D. dissertation, University of Calif., Davis, Sept. 1983.

[3] M. A. Soderstrand and B. Sinha, "Pipelined recursive RNS digital filter," *IEEE Trans. Circuits Syst.*, vol. 31, pp. 415–417, Apr. 1984.

[4] H. H. Loomis and B. Sinha, "High-speed recursive digital filter realization," *Circuits, Syst. Signal Process.*, vol. 3, pp. 267–294, Sept. 1984.

[5] M. A. Soderstrand, K. Chopper, and B. Sinha, "Comparison of three new techniques for pipelining IIR digital filters," in *Proc. 18th Asilomar Conf. Circuits, Syst. Computers.* Pacific Grove, CA: IEEE Computer Soc., Nov. 1984, pp. 439–443.

[6] H. Loomis, "Pipelined digital filters," U.S. Patent 4,495,591, Jan. 1985.

[7] K. K. Parhi and D. G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. 37, pp. 1099–1117, July 1989.

[8] M. A. Soderstrand, H. H. Loomis, and R. Gnanasekaran, "Pipelining techniques for IIR digital filters," in *Proc. 1990 IEEE Int. Symp. Circuits and Syst.*, New Orleans, LA: IEEE Circuits and Syst. Soc., May 1990, pp. 121–124.

[9] E. Q. Wong, M. A. Soderstrand, and H. H. Loomis, "Computer-aided design of pipelined IIR digital filters," in *Proc. 1992 IEEE Midwest Symp. Circuit Syst.*, Wash., DC, IEEE Circuits and Syst. Soc., Aug. 1992.

[10] H. Voelcker and E. Hartquist, "Digital filtering via block recursion," *IEEE Trans. Audio and Electroacoust.*, vol. 18, pp. 169–176, June 1970.

[11] A. de la Serna, "Stability of time-domain pipelined IIR digital filters," Master's thesis, Univ. California, Davis, CA, Sept. 1993.

[12] M. A. Soderstrand and A. E. de la Serna, "Stability of IIR filters constructed with time-domain pipelining technique," in *Proc. 1993 IEEE Midwest Symp. on Circuits Syst.*, Detroit, MI, IEEE Conf. Publ., Aug. 1993.

Design of FIR Filters with Complex Desired Frequency Response Using a Generalized Remez Algorithm

Michael Z. Komodromos, Steve F. Russell, and Ping Tak Peter Tang

***Abstract*—Complex approximation with a generalized Remez algorithm is used to design FIR digital filters with nonconjugate symmetric frequency responses. The minimax criterion is used and the Chebychev approximation is posed as a linear optimization problem. The primal problem is converted to its dual and is solved using an efficient, quadratically convergent algorithm developed by Tang [1]. Optimal Chebychev real-coefficient FIR filters with group delay smaller than half the filter length can be designed with slightly better magnitude responses compared to linear-phase filters. Linear-phase filters can also be designed when the group delay is specified to be half the filter length. Most importantly, the design method is capable to produce filters with complex coefficients that approximate nonconjugate symmetric frequency responses.**

I. Introduction

Most FIR filters are designed using the Parks-McClellan method [2], which is based on the real Remez algorithm [3]. Their widely used method produces conjugate-symmetric, linear-phase designs. Recently, the authors have investigated DSP filter applications, such as single-sideband filters in digital receivers, where it is desirable to relax the conjugate-symmetric and linear-phase restrictions. Our approach is to use constrained, weighted, complex approximation to compute the impulse response of an arbitrary FIR filter. Complex approximation allows specification of arbitrary filter magnitude and phase characteristics. In this paper we present a design method for FIR filters that can approximate a complex desired frequency response. The filter design problem is interpreted as a linear optimization problem. The primal problem is converted to its dual which is solved using an efficient algorithm based on the Generalized Remez Exchange Algorithm in [1].

Several methods have been suggested to design FIR filters in the complex domain. A method, developed by Chen and Parks [4], converts the primal complex approximation problem into a real approximation problem which is nearly equivalent to the complex problem. A standard linear programming algorithm is then used to solve the real approximation problem. The major disadvantage of this method is that the problem they solve is only an approximation (or discretization) to the original problem. As pointed out in [4], the execution time increases significantly as the discretization gets finer. Our method does not have this limitation. We will elaborate on this later.

Preuss [5] derived an interpolative design procedure similar to the real Remez algorithm. The algorithm deals directly with the complex error. In Preuss' algorithm, there appears to be no theoretical proof that the algorithm is convergent or, when it does, that the obtained approximation is optimal in some sense. Schulist [6], suggested a modified Preuss algorithm that improved the convergence of the algorithm.

Manuscript received December 14, 1992; revised January 4, 1994. This paper was recommended by Associate Editor Y. C. Lim.

M. Z. Komodromos and S. F. Russell are with the Department of Electrical Engineering and Computer Engineering, Iowa State University, Ames, Iowa 50011 USA.

P. T. P. Tang is with the Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 USA.

IEEE Log Number 9409269.

1057–7130/95$04.00 © 1995 IEEE

A recent paper by Alkhairy *et al.* [7] derives the dual of the primal approximation problem. The dual is solved using Tang's algorithm. The authors only consider the design of conjugate-symmetric filters which have real coefficients. Filters with Chebychev characteristics can also be designed using weighted least squares. An iterative method, developed by Lim *et al.* [8], derives the least squares frequency response weighting function that will produce a quasi-equiripple filter. The complexity of this method is $O(N^3)$ per iteration and as mentioned in [8], it converges to within 0.1 dB of the minimax solution within 6 iterations.

Our approach to designing FIR filters approximating arbitrary frequency responses will also be based on the complex approximation algorithm by Tang [1], [9] that is a generalization of the Remez algorithm for real approximation. This algorithm solves the dual formulation of the complex approximation problem without any need of discretization using a Simplex-like algorithm. Furthermore, it has been proven to quadratically converge to the best approximation. The complexity of the algorithm is approximately $O(N^3)$ per sweep, where a sweep corresponds to N Simplex iterations for a real-coefficient filter of length N. In the following sections we formulate the design of FIR filters in the complex domain as a minimization problem. We derive a common form for filters with real and complex coefficients. We then derive the dual equivalent maximization problem. The algorithm which solves the dual problem is presented next. Finally, design examples are presented.

II. Problem Formulation

We are required to approximate a complex-valued desired response, $D(z)$, with a realizable FIR filter transfer function, $H(z)$. The transfer function of an FIR filter with real impulse response is defined by the z-transform of its impulse response

$$H(z) = \sum_{k=0}^{n-1} h_k z^{-k} \tag{1}$$

where the frequency response, $H(f)$, is obtained by evaluating $H(z)$ on the unit circle, $z = e^{j2\pi f}$. In this paper we treat both cases of real and complex impulse responses. Even though the algorithm is applicable to both cases, the formulation of the two problems is unique to each case so we treat it separately. We examine first the case of real impulse response.

A. Real Impulse Response

Suppose the desired frequency response $D(z)$ is conjugate symmetric

$$D(z^*) = D^*(z) \tag{2}$$

The FIR filter design problem can be formulated as the following approximation problem. Find a set of optimal coefficients $\boldsymbol{h}' = (h'_0, \ldots, h'_{n-1}) \in \mathbb{C}^n$ such that

$$\| D(z) - H(\boldsymbol{h}', z) \| \leq \| D(z) - H(\boldsymbol{h}, z) \| \tag{3}$$

is satisfied for all $\boldsymbol{h} = (h_0, \ldots, h_{n-1}) \in \mathbb{C}^n$, where $\| \bullet \|$ denotes the Chebychev norm on the unit circle, and

$$H(\boldsymbol{h}, z) = \sum_{k=0}^{n-1} h_k b_k(z) \quad b_k(z) = z^{-k} \tag{4}$$

Because of the conjugate symmetry of the desired frequency response function, it can be proved that the optimal coefficients are real: $\boldsymbol{h} = (h_0, \ldots, h_{n-1}) \in \mathbb{R}^n$. Hence the impulse response is real. Moreover $\boldsymbol{h}'$ satisfies

$$\| D(z) - H(\boldsymbol{h}', z) \| \leq \| D(z) - H(\boldsymbol{h}, z) \| \tag{5}$$

for all $\boldsymbol{h} = (h_0, \ldots, h_{n-1}) \in \mathbb{R}^n$, where the Chebychev norm is taken on the unit circle on the upper half plane. The upper half unit circle corresponds to the normalized frequency range [0, 0.5], which is the domain of approximation.

For the design of linear-phase filters, we take advantage of the symmetry of the impulse response coefficients. An odd order linear-phase approximating FIR filter and the basis functions are defined as follows:

$$H(f) = \sum_{k=0}^{n-\frac{1}{2}} h_k b_k(f), b_k(f) = e^{-j2\pi fk} + e^{-j2\pi f(N-1-k)} \tag{6}$$

B. Complex Impulse Response

When it is required to approximate a nonconjugate symmetric frequency response, the impulse response of the approximating FIR filter must be complex. The transfer function of an FIR filter with complex impulse response is given by

$$H(z) = \sum_{k=0}^{n-1} h_{kc} z^{-n} \tag{7}$$

where the coefficients h_{kc}, $k = 0, \ldots, n-1$ are the complex impulse response, f is the normalized frequency in the interval [-0.5, 0.5], and n is the length of the filter. We can break the complex coefficients into their real and imaginary parts and define them as follows:

$$h_{kc} = h_k + j h_{n+k} \quad k = 0 \cdots n-1 \tag{8}$$

Then we write the transfer function as

$$H(z) = \sum_{k=0}^{n-1} \left[h_k(z^{-k}) + h_{n+k}(j z^{-k}) \right] = \sum_{k=0}^{2n-1} h_k b_k(z) \tag{9}$$

where

$$b_k(z) = z^{-k} \quad b_{k+n}(z) = j z^{-k} \quad k = 0, \ldots, n-1 \tag{10}$$

are the basis functions. We can treat the real and complex impulse response problems simultaneously by replacing the upper index n by N and noting that for the real case $N = n$ and for the complex case $N = 2n$. Also the basis functions for the real case are defined in (4) and for the complex case in (10). The frequency response of the FIR filter can be obtained by evaluating the z-transform on the unit circle which results in

$$H(f) = \sum_{k=0}^{N-1} h_k b_k(f) \tag{11}$$

C. Primal Problem

The complex filter approximation problem is stated as the following primal problem. Let $D(f)$ denote the desired complex-valued frequency response on F, a finite set of disjoint continuous intervals in [-0.5, 0.5] for the complex impulse response case, and [0, 0.5] for the real case, and let $H(f)$ denote the approximating FIR filter response. Define the weighted complex error between the desired response and the approximating FIR filter by

$$E(f) = W(f)[D(f) - H(f)] \quad f \in F \tag{12}$$

where $W(f)$ is a positive real-valued weighting function introduced to control the relative magnitude of the complex error in the different frequency bands. Find N real parameters $\boldsymbol{h}' = [h'_0, h'_1, \ldots, h'_{N-1}]$

such that

$$\max_{f \in F} \left\{ W(f) \mid D(f) - \sum_{k=0}^{N-1} h'_k b_k(f) \mid \right\} \leq \max_{f \in F} \left\{ W(f) \mid D(f) - \sum_{k=0}^{N-1} h_k b_k(f) \mid \right\} \quad (13)$$

for all $\boldsymbol{h} = [h_0, h_1, \ldots, h_{N-1}]^T$ in $\mathbb{R}^N$. The error is smaller in the frequency bands with larger weight. Since the weight function is positive we incorporate it with the desired response and the approximating function.

An equivalent formulation of the approximation problem is to find optimal variables $\delta, h_1, \ldots, h_{N-1}$ to minimize the linear function δ subject to the constraints

$$\mid D(f) - \sum_{k=0}^{N-1} h_k b_k(f) \mid \leq \delta \quad (14)$$

for all $f \in F$. Note that although the function to be minimized is linear with respect to the variables, the constraints are not. The latter can be made linear by exploiting the fact that for any complex number z

$$\mid z \mid = \max_{\alpha \in [0, 2\pi]} \left\{ Re\{z e^{-j\alpha}\} \right\} \quad (15)$$

Thus the constraints are equivalent to

$$\sum_{k=0}^{N-1} h_k Re\{b_k(f) e^{-j\alpha}\} + \delta \geq Re\{D(f) e^{-j\alpha}\} \quad (16)$$

for all $f \in F$ and for all $\alpha \in [0, 2\pi]$. If the sets F and $[0, 2\pi]$ were replaced by two finite sets, the problem reduces to a standard linear programming problem, the solution of which can be obtained by applying the well-known Simplex method. But then, standard theory there suggests that we should solve the dual of this problem instead because it is much more efficient to tackle a problem with fewer constraints than variables. In the following section, we formulate the dual and show there is, in fact, no need to replace F or $[0, 2\pi]$ by finite sets.

D. Dual Problem

The primal approximation problem is converted to its dual equivalent because i) the number of constraints becomes finite and ii) the dual makes it is easier to derive an efficient algorithm, and iii) the results are theoretically equivalent, i.e., no discretization for continuous frequency or angle is required. The dual of the primal problem is defined as follows [1], [9], [10]. Define the $(N+1)x1$ vectors

$$\boldsymbol{f} \in F^{N+1}, \boldsymbol{\alpha} \in [0, 2\pi]^{N+1}, \boldsymbol{r} \in [0, 1]^{N+1} \quad (17)$$

Find the vectors $\boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{r}$, to maximize the dual variable d,

$$d = c^T \cdot \boldsymbol{r} \quad (18)$$

subject to the constraints

$$\boldsymbol{A} \cdot \boldsymbol{r} = \boldsymbol{u_1} \quad (19)$$

where

$$\boldsymbol{c}^T \left[Re\{D(f_1) e^{-j\alpha_k}\} \cdots Re\{D(f_{N+1}) e^{-j\alpha_{N+1}}\} \right]^T \quad (20)$$

and the kth column of the $(N+1)x(N+1)$ matrix $\boldsymbol{A}$ is

$$\left[1 \;\; Re\{b_0(f_k) e^{-j\alpha_k}\} \cdots Re\{b_{N-1}(f_k) e^{-j\alpha_k}\} \right]^T \quad (21)$$

The vector $\boldsymbol{u_1}$ is the $(N+1 \; x \; 1)$ coordinate column vector with the first element of one and the rest zero.

The relation between the primal and dual problems is that the optimal (minimized) error of the primal is the same as the optimal (maximized) error of the dual. Furthermore, there is a relation among the optimal values of $d, \boldsymbol{h}$ with the extremal points $\boldsymbol{f}, \boldsymbol{\alpha}$ which allows the calculation of the maximum error and the impulse response of the filter. This relation will be further described when the algorithm is explained in the next section.

The approximating functions must satisfy some constraints to qualify as basis functions for the complex approximation. These conditions are required to guarantee uniqueness of the approximation. The basis functions have to be linearly independent on their domain F and also satisfy the Haar condition [11]. Existence and uniqueness of the best complex approximation is established in [11].

III. Complex Remez Algorithm

The design algorithm solves the dual problem described in the previous section. The following theorem adapted from Tang [1], [9] suggests the algorithm.

Theorem: Suppose $H(\boldsymbol{h}', \bullet)$ is a best approximation to a complex-valued desired frequency response $D(f)$, and let $d' \equiv \| E(\boldsymbol{h}', \bullet) \|$ be the error of the best approximation. Then d' is the optimal value of the following constrained maximization problem: Maximize the inner product $d = \boldsymbol{c}^T \cdot \boldsymbol{r}$ over all $\boldsymbol{f} \in F^{N+1}$, $\boldsymbol{\alpha} \in [0, 2\pi]^{N+1}$ and $\boldsymbol{r} \in [0, 1]^{N+1}$ subject to the constraints

$$\boldsymbol{A} \cdot \boldsymbol{r} = u_1 \quad (22)$$

The theorem suggests the following algorithm for the determination of the best FIR filter:

Step 1: Pick a set of initial frequencies and angles $(\boldsymbol{f}, \boldsymbol{\alpha})$ such that

$$\boldsymbol{r} = \boldsymbol{A}^{-1} \cdot \boldsymbol{u_1} \geq \mathbf{o} \quad (23)$$

The initial set has to be chosen in a way that the matrix $\boldsymbol{A}$ is nonsingular so its inverse exists. This is usually the case when the initial set is chosen at random. From $(\boldsymbol{f}, \boldsymbol{\alpha})$, calculate the corresponding inner product d and a set of coefficients $\boldsymbol{h}$.

Step 2: Terminate the algorithm if the value of the inner product $d = \boldsymbol{c}^T \cdot \boldsymbol{r}$ is close enough to optimal. By this we mean that d is within a certain tolerance factor to $\| E(\boldsymbol{h}, \bullet) \|$, the Chebychev norm of the complex error.

Step 3: Update the extremal set $(\boldsymbol{f}, \boldsymbol{\alpha})$ and the weight vector $\boldsymbol{r}$ (without violating any of the constraints) to increase the value of the scalar product $d = \boldsymbol{c}^T \cdot \boldsymbol{r}$. This is the exchange step in the algorithm. Return to step 2 and repeat the process until the algorithm terminates.

Let us now discuss the steps of the algorithm in more detail. If the inverse of the resulting $\boldsymbol{A}$ is ill-conditioned a different random starting point is chosen until a satisfactory $\boldsymbol{A}$ is found. In practice, the first matrix $\boldsymbol{A}$ is usually sufficiently well-conditioned; we never need to go beyond the second random matrix. Since $\boldsymbol{A}$ is invertible, this will be our initial matrix provided

$$\boldsymbol{r} = \boldsymbol{A}^{-1} \cdot \boldsymbol{u}_1 \geq \mathbf{o} \quad (24)$$

If that is not the case, we adjust α by $\alpha_k \rightarrow \alpha_k + \pi$ whenever $r_k < 0$. One can prove that if the resulting new $\boldsymbol{A}$ remains invertible, we must have

$$\boldsymbol{r} = \boldsymbol{A}^{-1} \cdot \boldsymbol{u}_1 \geq \mathbf{o} \quad (25)$$

Although there is no theoretical guarantee that the adjusted $\boldsymbol{A}$ must be invertible, we never encounter problems in this adjustment in practice. This method of finding an initial $\boldsymbol{A}$ is rather natural and was also used

in the code published in [9]. In Step 2, the value of d is checked for optimality. It can be shown that the following relations hold [9], [10]:

$$[d, \boldsymbol{h}]^T \cdot \boldsymbol{A} = c^T \tag{26}$$

and [9]

$$d \leq \| E(\boldsymbol{h}', \bullet) \| \leq \| E(\boldsymbol{h}, \bullet) \| \tag{27}$$

These equations suggest the following stopping criterion of the algorithm. The value d is calculated at each iteration and it is compared with $\| E(\boldsymbol{h}, \bullet) \|$. If

$$\| E(\boldsymbol{h}, \bullet) \| - d \leq tolerance\ d \tag{28}$$

for some "tolerance" chosen beforehand, the algorithm can be terminated. A termination with tolerance chosen as 0.01 means the approximation obtained is within 1% of the truly optimal one. Clearly, a smaller tolerance can be specified if a closer approximation to the best is desired.

The one point exchange in Step 3 is performed on the frequency points and angles (extremal signature) of the current iteration until optimality has been reached. First the error is calculated on the domain of approximation and the frequency f_{new}, at which the maximum magnitude of the error is attained, is located. This process does not necessarily require a discretization of the frequency domain F because we can represent the derivative of the magnitude of the error in closed form and a zero finder can be applied. A discussion on the derivative of the magnitude of the error can be found in [10], and an implementation using a zero-finder can be found in [9]. After f_{new} is determined, we determine α_{new}, the phase angle of the complex error at f_{new}. Hence, there is no need to discretize the phase domain of $[0, 2\pi]$. This pair (f_{new}, α_{new}) is then used to replace an appropriate pair in the old extremal set.

The exchange of the extremal set should result in an increase of the dual variable d. The set (f_m, α_m) to be replaced is found as follows. Construct the vector $\boldsymbol{v}$, where $\boldsymbol{v}$ is the same as one of the columns of $\boldsymbol{A}$ at $(f)_{new}, \alpha_{new}$. Then, define the vector $\boldsymbol{p} = \boldsymbol{A}^{-1} \cdot \boldsymbol{v}$. It can be proved [9] that some of the elements of $\boldsymbol{p}$ are positive. The set (f_m, α_m) is chosen by finding an index m, $1 \leq m \leq N+1$, , such that i) $p_m > 0$, and ii) $r_m/p_m \leq r_j/p_j$ when $p_j > 0$. This method has been used by Tang [9] and it is also the standard method for the Simplex algorithm [12]. When the pivotal point is found and the exchange is performed, the algorithm returns to Step 2 and repeats.

The complexity of Step 1 is $0(N^3)$. This includes the first LU factorization and solution of the initial basis matrix to get the primal and dual vectors. Each iteration involves two main computational processes: i) location of the maximum magnitude of the error function, and ii) update of the basis and primal and dual solutions. For i), if the function evaluation at each frequency point is done in a straightforward manner the complexity is $O(N)$. If a discretized frequency domain is used, and the maximum magnitude is determined by evaluation of the magnitude at every single frequency point in a domain of M points, say $M = 4N$, then the complexity is $O(4N^2)$. Thus, for one sweep of N iterations, the complexity is $O(4N^3)$. For ii), if the updating is done by solving for the primal and dual variables with the new basis matrix, the complexity is $O(N^3)$. However, a simplification is possible by exploiting the fact that the new basis matrix differs from the previous one by only one column. In Tang [9], the implementation uses an $O(N^2)$updating scheme which was based on well-known rank-one updating methods in computational linear algebra. An $O(N^2)$ approach based on the Shermann-Morrison formula is also mentioned by Alkhairy *et al.* [7], although no details were given. Thus, for one sweep, the complexity is $O(N^3)$. To be numerically safe, in practice, we recompute the LU factorization of the basis matrix after every sweep, adding another $\frac{1}{3}N^3$ operations at every sweep. Summarizing then, every sweep requires less than $(5N^2)$.

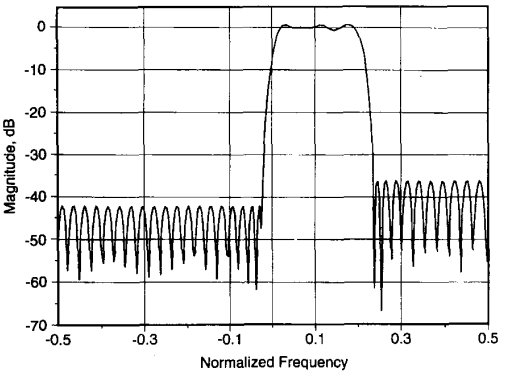

Fig. 1. Magnitude plot, in dB, for the complex filter of example 2.

In our experience, to obtain a solution within 1% of the true best approximation, 4 to 5 sweeps suffice. A closer solution to the optimal is usually achieved between 6-10 sweeps. One more note needs to be made concerning the location of the maximum magnitude of the error. When the frequency domain is a continuum (as opposed to discretized frequency points), one can locate the maximum by solving for the zeros of the derivative of the magnitude of the error. In Tang [9], a comparison on the number of error function evaluations needed between using a zero finder and a straightforward sampling was performed. The number of function evaluations required using a zero finder can be three or four times fewer. We have stayed with a discretized frequency, however, for the purpose of better comparison with existing work [4], [5], [7], and [8].

IV. Design Examples

The FIR filter design method was implemented with programs written in FORTRAN and MATLAB. The input to the programs include the normalized frequency band edges, the value of the magnitude in each band, the weight for each band, and the desired group delay in each band.

Example 1: This example from Chen [4] is a lowpass design of length $N = 31$, passband [0, 0.06] and stopband [0.12, 0.5]. The group delay is specified to be 12 samples and a weight ratio 1:10 is used. We use the discretization used in [7], with actual passband frequency 61/1024 and stopband frequency 123/1024 for comparison. The Chebychev error within 0.01% of the optimal value is 0.0422 and was accomplished in 10 sweeps. This results in a passband ripple of $\delta_p = 0.0422$ (0.36 dB) and stopband attenuation of $\delta_p = 0.00422$ (47.49 dB). The Chebychev error was 0.0436 in 4 sweeps and 0.0424 in 6 sweeps. The passband group delay deviation from the specified 12 samples is approximately 1 sample. A lowpass filter design with the same cutoff frequencies with length $N = 81$ and specified group delay 32 samples and weight ratio 1:10 gives a passband ripple $\delta_p =$ 0.0 003154 (0.00274 dB), and stopband attenuation of $\delta_s = 0.00003$ 154 (90.0 dB). This example converged within 0.01% of the best approximation in 7 sweeps.

Example 2: This example is an approximation to a nonconjugate symmetric frequency response, resulting in an FIR filter with complex coefficients. The length of the filter is 37 (complex) with stopbands [-0.5,-0.02] and [0.24, 0.5] and passband [0.02, 0.2]. The desired group delay is τ_d=14 samples and the weight ratio is 10:1:5. The algorithm converged in 9 sweeps within 0.1% of the best solution. The magnitude in dB is shown in Fig. 1 with passband ripple

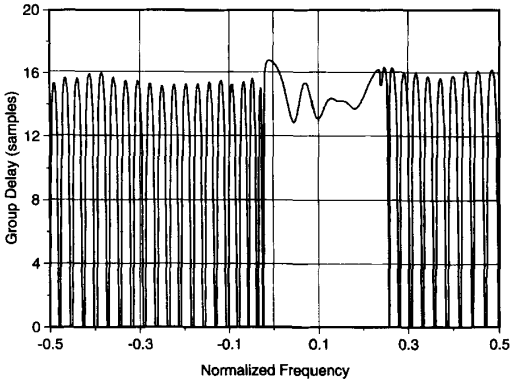

Fig. 2. Group delay plot for the complex filter of example 2.

δ_p = 0.07766 (0.65 dB), and attenuation in the first stopband δ_{s1} =0.007766 (42.20 dB), and attenuation in the second stopband δ_{s2} =0.01553 (36.16 dB). The filter group delay is shown in Fig. 2. The maximum group delay in the passband is 15.52 samples and the minimum is 12.81 samples. When a lower group delay is specified the Chebychev error and maximum group delay deviation are slightly poorer. The best Chebychev error occurs when a group delay of 18 samples is specified, which corresponds to the linear phase case.

V. Conclusion

This paper presented a new design approach for the design of FIR filters approximating an arbitrary complex-valued desired frequency response. The linear-phase restriction of the Parks-McClellan method has been removed. More importantly, the conjugate symmetry restriction is also removed. This gives enormous freedom in specifying arbitrary magnitude and phase responses. The filters are designed using a minimax criterion to minimize the Chebychev error between the complex desired frequency response and the FIR frequency response. The complex approximation problem is considered as a linear optimization problem. The dual of the primal problem was derived because it results in an efficient algorithm for the determination of the optimal complex filter in the Chebychev sense. The algorithm produces good nearly linear-phase filters with group delays other than one-half of the filter length. It also produces linear-phase filters that are the same as those obtained by the Parks-McClellan algorithm for half-length delays. Finally, FIR filters approximating nonconjugate symmetric responses can be designed. This problem was shown to be an extension of the approximation problem with real coefficients.

References

[1] P. T. P. Tang, "A fast algorithm for linear complex Chebychev approximations," *Math. Computat.*, vol. 51, no. 184, pp. 721–739, Oct. 1988.

[2] J. H. McClellan, T. W. Parks, and L.R. Rabiner, "A computer program for designing optimum FIR linear phase digital filters," *IEEE Trans. Audio and Electroacoust.*, vol. AU-21, no. 6 pp. 506–526, Dec. 1973.

[3] E. Ya. Remez, *General Computational Methods of Chebychev Approximation*. Kiev, USSR: Atomic Energy Translation 4491, 1957.

[4] X. Chen and T. W. Parks, "Design of FIR filters in the complex domain," *IEEE Trans. ASSP*, vol. ASSP-35, no. 2, pp. 144-153, Feb. 1987.

[5] K. Preuss, "On the design of FIR filters by complex Chebychev approximation," *IEEE Trans. ASSP*, vol. 37, no. 5, pp. 702–712, May 1989.

[6] M. Schulist, "Improvements of a complex FIR filter design algorithm," *Signal Process.*, vol. 20, no. 1, pp. 81-90, May 1990.

[7] A. Alkhairy, K. G. Christian, and J. Lim, "Design and characterization of optimal FIR filters with arbitrary phase," *IEEE Trans. Signal Process.*, vol. 41, no. 2, pp. 559–572, Feb. 1993.

[8] Y. Lim, J. Lee, C. Chen, and R. Yang, "A weighted least squares algorithm for quasi-equiripple FIR and IIR digital filter design," *IEEE Trans. Signal Process.*, vol. 40, no. 3, pp. 551–558, Mar. 1992.

[9] P. T. P. Tang, "Chebychev approximation on the complex plane," Ph.D. dissertation, Dept. Mathematics, Univ. California at Berkeley, May 1987.

[10] M. Z. Komodromos, "Optimal FIR filter design," Ph.D. dissertattion, Dept. Elect. Eng. and Computer Eng., Iowa State Univ., Dec. 1993.

[11] G. Meinardus, *Approximations of Functions: Theory and Numerical Methods*, (translated by L. L. Schumaker). New York: Springer-Verlag, 1967.

[12] D. G. Luenberger, *Linear and Nonlinear Programming*, second ed. Reading, MA: Addison-Wesley, 1984.

Scaled Normalized Lattice Digital Filter Structures

Jin-Gyun Chung and Keshab K. Parhi

***Abstract*—The minimum-noise lattice digital filter with a systolizable structure, good scaling property and low output roundoff noise was recently proposed. In this paper, we show that the same structure can be designed by applying a new scaling method to the well known normalized lattice digital filter. Therefore, this structure can be referred to as the scaled normalized lattice digital filter. Different types of scaled normalized lattice digital filter structures are also derived.**

I. Introduction

When a digital filter transfer function is implemented using a digital system, it invariably involves quantization of signals and coefficients in the system. As a result, the overall input-output behavior is not ideal. Therefore, much research has been carried out to search for low-sensitivity or low-noise structures such as *wave digital filters* [1], *digital lattice or ladder filters* [2], [3], *double-rotation lattice digital filters* [4] and *minimum-noise lattice filters* [5]. In addition to finite word-length effects, pipelining is very important in low-power or high sample rate applications. Pipelining methods of the above mentioned filter structures have been proposed recently in [6–8].

The minimum-noise lattice filter is designed using the techniques of evaluating both the internal signal powers and output roundoff noise based on the Schur algorithm [9], the polynomial expansion algorithm, Parseval's theorem, and a local optimization approach. In this paper, we show that the minimum-noise lattice filter structure can be obtained by just applying a new scaling method to the well-known normalized lattice filter in [3]. The scaling method in [10] is based on a state variable description. However, lattice filters generally

Manuscript received October 9, 1992; revised August 13, 1994. This work was supported by the Army Research office under Contract DAAL-90-G-0063 and the National Science Foundation under Contract MIPS-9258670. This paper was recommended by Associate Editor Y. C. Lim.

The authors are with the Department of Electrical Engineering, University of Minnesota, Minneapolis, MN 55455 USA.

IEEE Log Number 9409259.

1057–7130/95$04.00 © 1995 IEEE